

Einladung

zum Informatik-Kolloquium des
AB Programmiersprachen und Übersetzer am
Mittwoch, den 25. Mai 2011, um 15 Uhr c.t.
im HS 8 Heinz Parkus, Hauptgebäude, Karlsplatz 13 (Stiege 7)

Es spricht

Ass.Prof. Dr. Matthias Hauswirth

University of Lugano, Switzerland

über

Software: Design and Performance

This talk consists of two parts: the first part, based on our ECOOP'11 paper on “essence”, focuses on software design, and the second part, based on our recently launched ParaBoost project, focuses on software performance.

In the first part of this talk we present an approach that partitions a software system into its algorithmically essential parts and the parts that manifest its design. Our approach is inspired by the notion of an algorithm and its asymptotic complexity. However, we do not propose a metric for measuring asymptotic complexity (efficiency). Instead, we use the one aspect of algorithms that drives up their asymptotic complexity – repetition, in the form of loops and recursions – to determine the algorithmically essential parts of a software system. Those parts of a system that are not algorithmically essential represent aspects of the design. A large fraction of inessential parts is indicative of “overdesign”, while a small fraction indicates a lack of modularization. We present a metric, relative essence, to quantify the fraction of the program that is algorithmically essential. We evaluate our approach by studying the algorithmic essence of a large corpus of software systems, and by comparing the measured essence to an intuitive view of design “overhead”.

In the second part of the talk we present the recently launched ParaBoost project in which we are exploring approaches to benefit from the ever increasing core count in modern processors without requiring the explicit parallelization of applications. The project is based on the existing idea of competitive parallel execution (CPE). CPE lets multiple algorithms compete, each on its own core, on solving the same problem. The algorithm that finds the solution first wins the competition. The competition thus takes only as long as the fastest concurrently executing algorithm. In the ParaBoost project, we combine this idea of concurrently competing algorithm variants with the software engineering benefits afforded by modern programming languages. We are creating a managed runtime environment called a multi-variant virtual machine (MVVM), to enable existing Java software to benefit from multi-variant execution. By lifting multi-variant execution into the language runtime, we will gain fine-grained control over the execution of the variants. This control will allow us to go beyond existing CPE approaches: The MVVM will be able to competitively execute software that uses more than one thread, it will improve performance by dynamically allocating more time to more promising variants, and it will learn and use models predicting the performance of variants based on features of their input.

Biography: Matthias Hauswirth is an assistant professor at the Faculty of Informatics at the University of Lugano (Switzerland), where he leads the Software and Programmer Efficiency (Sape) research group. He is interested in performance measurement, understanding, and optimization. Matthias received his PhD from the University of Colorado at Boulder. (<http://www.inf.usi.ch/faculty/hauswirth/> / <http://sape.inf.usi.ch/>)

Tee: 14:30 Uhr in der Bibliothek E185.1, Argentinierstr. 8, 4. Stock (Mitte).